

UNIVERSIDADE ESTADUAL DE SANTA CRUZ

CET058 - COMPILADORES

Leonardo Damasceno Nascimento - 201810368

Projeto 1 B: P-code Machine

Ilhéus/BA 2022

Sumário

3. Implementação

4. Exemplo de códigos:

- 4. Soma de dois números inteiros (p-code e saída do código)
- 4. Soma dos números inteiros naturais de 1 até 100 ((p-code)
- 5. Soma dos números inteiros naturais de 1 até 100 ((saída do código)
- 6. Soma dos números naturais de 1 até 100 (p-code e saída do código)
- 7. Soma dos quadrados dos números naturais de 1 até 100 (p-code e saída do código)
- 8. Soma dos Cubos dos números naturais de 1 até 100 (p-code e saída do código)
- 9. Download do Código
- 9. Referências

IMPLEMENTAÇÃO

Este relatório tem como o objetivo demonstrar um código que simula um leitor de código p-code e executa dentro da máquina/código.

Esse simulador tem como referência o código presente no artigo da wikipedia ([link](#)).

Esse código base é desenvolvido em pascal para que simule uma máquina virtual para rodar o portable code machine (P-code). A tarefa foi fazer o mesmo, porém com a linguagem C.

Linguagem Utilizada: C

Inputs permitidos: Serão permitidos arquivos .txt que apresentarem itens compatíveis com a linguagem do P-code que estão pré-definidos como:

- LIT;
- OPR ;
- LOD ;
- STO ;
- CAL ;
- INT ;
- JMP;
- JPC ;

Com os exemplos do código ficará melhor a visualização do funcionamento e disposição das informações.

Tratativa de erros: Ao inserir um .txt que não seja compatível ele irá rodar, porém não executará da maneira desejável. Apresentando no terminal de saída:

```
start pl/0
t  b  p      f  l  a          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
=== === ===  === === ===    === === === === === === === === ===
-1  0  0  OPR  0  0      s[] :  === === === === === === === === ===
=== === ===  === === ===    === === === === === === === === ===
t  b  p      f  l  a          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

Exemplos de códigos:

1. Soma de dois números inteiros:

a. P-code:

```
1  INT 0 4
2  LIT 0 2
3  LIT 0 3
4  OPR 0 2
5  STO 0 3
6  OPR 0 0
```

b. Saída do código:

```
t  b  p  f  l  a      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
=== ===
-1  0  0  INT  0  4    s[] :  0  0  0  0
 3  0  1  LIT  0  2    s[] :  0  0  0  0  2
 4  0  2  LIT  0  3    s[] :  0  0  0  0  2  3
 5  0  3  OPR  0  2    s[] :  0  0  0  0  5
 4  0  4  STO  0  3    s[] :  0  0  0  5
 3  0  5  OPR  0  0    s[] :
=== ===
t  b  p  f  l  a      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
end pl/0
```

2. Soma dos números inteiros naturais de 1 até 100 (iterativamente)

a. P-code:

```
1  INT 0 4    16  OPR 0 2
2  LIT 0 1    17  STO 0 3
3  LIT 0 2    18  LOD 0 3
4  OPR 0 2    19  LIT 0 6
5  STO 0 3    20  OPR 0 2
6  LOD 0 3    21  STO 0 3
7  LIT 0 3    22  LOD 0 3
8  OPR 0 2    23  LIT 0 7
9  STO 0 3    24  OPR 0 2
10 LOD 0 3    25  STO 0 3
11 LIT 0 4    26  LOD 0 3
12 OPR 0 2    27  LIT 0 8
13 STO 0 3    28  OPR 0 2
14 LOD 0 3    29  STO 0 3
15 LIT 0 5    30  LOD 0 3
31  LIT 0 9
32  OPR 0 2
33  STO 0 3
34  LOD 0 3
35  LIT 0 10
36  OPR 0 2
37  STO 0 3
38  OPR 0 0
```

b. Saída do código:

```
start p1/0

t   b   p   f   l   a           s[] :   0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
=== === ===   === === ===   === ===
-1  0   0   INT  0   4   s[] :   0   0   0   0
 3  0   1   LIT  0   1   s[] :   0   0   0   0   1
 4  0   2   LIT  0   2   s[] :   0   0   0   0   1   2
 5  0   3   OPR  0   2   s[] :   0   0   0   0   3
 4  0   4   STO  0   3   s[] :   0   0   0   3
 3  0   5   LOD  0   3   s[] :   0   0   0   3   3
 4  0   6   LIT  0   3   s[] :   0   0   0   3   3   3
 5  0   7   OPR  0   2   s[] :   0   0   0   3   6
 4  0   8   STO  0   3   s[] :   0   0   0   6
 3  0   9   LOD  0   3   s[] :   0   0   0   6   6
 4  0  10   LIT  0   4   s[] :   0   0   0   6   6   4
 5  0  11   OPR  0   2   s[] :   0   0   0   6  10
 4  0  12   STO  0   3   s[] :   0   0   0  10
 3  0  13   LOD  0   3   s[] :   0   0   0  10  10
 4  0  14   LIT  0   5   s[] :   0   0   0  10  10   5
 5  0  15   OPR  0   2   s[] :   0   0   0  10  15
 4  0  16   STO  0   3   s[] :   0   0   0  15
 3  0  17   LOD  0   3   s[] :   0   0   0  15  15
 4  0  18   LIT  0   6   s[] :   0   0   0  15  15   6
 5  0  19   OPR  0   2   s[] :   0   0   0  15  21
 4  0  20   STO  0   3   s[] :   0   0   0  21
 3  0  21   LOD  0   3   s[] :   0   0   0  21  21
 4  0  22   LIT  0   7   s[] :   0   0   0  21  21   7
 5  0  23   OPR  0   2   s[] :   0   0   0  21  28
 4  0  24   STO  0   3   s[] :   0   0   0  28
 3  0  25   LOD  0   3   s[] :   0   0   0  28  28
 4  0  26   LIT  0   8   s[] :   0   0   0  28  28   8
 5  0  27   OPR  0   2   s[] :   0   0   0  28  36
 4  0  28   STO  0   3   s[] :   0   0   0  36
 3  0  29   LOD  0   3   s[] :   0   0   0  36  36
 4  0  30   LIT  0   9   s[] :   0   0   0  36  36   9
 5  0  31   OPR  0   2   s[] :   0   0   0  36  45
 4  0  32   STO  0   3   s[] :   0   0   0  45
 3  0  33   LOD  0   3   s[] :   0   0   0  45  45
 4  0  34   LIT  0  10   s[] :   0   0   0  45  45  10
 5  0  35   OPR  0   2   s[] :   0   0   0  45  55
 4  0  36   STO  0   3   s[] :   0   0   0  55
 3  0  37   OPR  0   0   s[] :

=== === ===   === === ===   === ===
t   b   p   f   l   a           0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
end p1/0
```

3. Soma dos números naturais de 1 até 100 (iterativamente)

a. P-code:

| | | | | | | | | | | | | | |
|---|-----|---|-----|----|-----|---|----|--|--|----|-----|---|---|
| 1 | INT | 0 | 6 | | | | | | | 15 | LOD | 0 | 3 |
| 2 | LIT | 0 | 1 | | | | | | | 16 | OPR | 0 | 2 |
| 3 | STO | 0 | 3 | 9 | INT | 0 | -1 | | | 17 | STO | 0 | 4 |
| 4 | LIT | 0 | 0 | 10 | LOD | 0 | 3 | | | 18 | LOD | 0 | 3 |
| 5 | STO | 0 | 4 | 11 | LOD | 0 | 5 | | | 19 | LIT | 0 | 1 |
| 6 | LIT | 0 | 100 | 12 | OPR | 0 | 12 | | | 20 | OPR | 0 | 2 |
| 7 | STO | 0 | 5 | 13 | JPC | 0 | 22 | | | 21 | STO | 0 | 3 |
| 8 | LIT | 0 | 0 | 14 | LOD | 0 | 4 | | | 22 | JMP | 0 | 8 |
| 9 | INT | 0 | -1 | 15 | LOD | 0 | 3 | | | 23 | OPR | 0 | 0 |

b. Saída do código:

A partir deste momento o código fica extremamente grande para colocar em imagens, então colocarei apenas o final.

```

6  0 12 JPC 0 22 s[] : 0 0 0 100 4950 100 0
6  0 13 LOD 0 4 s[] : 0 0 0 100 4950 100 0 4950
7  0 14 LOD 0 3 s[] : 0 0 0 100 4950 100 0 4950 100
8  0 15 OPR 0 2 s[] : 0 0 0 100 4950 100 0 5050
7  0 16 STO 0 4 s[] : 0 0 0 100 5050 100 0
6  0 17 LOD 0 3 s[] : 0 0 0 100 5050 100 0 100
7  0 18 LIT 0 1 s[] : 0 0 0 100 5050 100 0 100 1
8  0 19 OPR 0 2 s[] : 0 0 0 100 5050 100 0 101
7  0 20 STO 0 3 s[] : 0 0 0 101 5050 100 0
6  0 21 JMP 0 8 s[] : 0 0 0 101 5050 100 0
6  0 8 INT 0 -1 s[] : 0 0 0 101 5050 100
5  0 9 LOD 0 3 s[] : 0 0 0 101 5050 100 101
6  0 10 LOD 0 5 s[] : 0 0 0 101 5050 100 101 100
7  0 11 OPR 0 12 s[] : 0 0 0 101 5050 100 1
6  0 12 JPC 0 22 s[] : 0 0 0 101 5050 100
5  0 22 OPR 0 0 s[] :
=== === ===
t b p f l a 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
end pl/0

```

4. Soma dos quadrados dos números naturais de 1 até 100 (iterativamente)

a. P-code:

| | | | |
|----|-----------|----|----------|
| 1 | INT 0 6 | 11 | LOD 0 5 |
| 2 | LIT 0 1 | 12 | OPR 0 12 |
| 3 | STO 0 3 | 13 | JPC 0 24 |
| 4 | LIT 0 0 | 14 | LOD 0 3 |
| 5 | STO 0 4 | 15 | LOD 0 3 |
| 6 | LIT 0 100 | 16 | OPR 0 4 |
| 7 | STO 0 5 | 17 | LOD 0 4 |
| 8 | LIT 0 0 | 18 | OPR 0 2 |
| 9 | INT 0 -1 | 19 | STO 0 4 |
| 10 | LOD 0 3 | 20 | LOD 0 3 |

b. Saída do código:

| t | b | p | f | l | a | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-------|---|---|---|-----|--------|-----|----------------|---|---|---|----|----|----|----|----|
| 6 | 0 | 12 | JPC | 0 | 24 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 | | | | | | | | |
| 6 | 0 | 13 | LOD | 0 | 3 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 100 | | | | | | | | |
| 7 | 0 | 14 | LOD | 0 | 3 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 100 100 | | | | | | | | |
| 8 | 0 | 15 | OPR | 0 | 4 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 10000 | | | | | | | | |
| 7 | 0 | 16 | LOD | 0 | 4 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 10000 328350 | | | | | | | | |
| 8 | 0 | 17 | OPR | 0 | 2 | s[] : | 0 | 0 | 0 | 100 | 328350 | 100 | 0 338350 | | | | | | | | |
| 7 | 0 | 18 | STO | 0 | 4 | s[] : | 0 | 0 | 0 | 100 | 338350 | 100 | 0 | | | | | | | | |
| 6 | 0 | 19 | LOD | 0 | 3 | s[] : | 0 | 0 | 0 | 100 | 338350 | 100 | 0 100 | | | | | | | | |
| 7 | 0 | 20 | LIT | 0 | 1 | s[] : | 0 | 0 | 0 | 100 | 338350 | 100 | 0 100 1 | | | | | | | | |
| 8 | 0 | 21 | OPR | 0 | 2 | s[] : | 0 | 0 | 0 | 100 | 338350 | 100 | 0 101 | | | | | | | | |
| 7 | 0 | 22 | STO | 0 | 3 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | 0 | | | | | | | | |
| 6 | 0 | 23 | JMP | 0 | 8 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | 0 | | | | | | | | |
| 6 | 0 | 8 | INT | 0 | -1 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | | | | | | | | | |
| 5 | 0 | 9 | LOD | 0 | 3 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | 101 | | | | | | | | |
| 6 | 0 | 10 | LOD | 0 | 5 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | 101 100 | | | | | | | | |
| 7 | 0 | 11 | OPR | 0 | 12 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | 1 | | | | | | | | |
| 6 | 0 | 12 | JPC | 0 | 24 | s[] : | 0 | 0 | 0 | 101 | 338350 | 100 | | | | | | | | | |
| 5 | 0 | 24 | OPR | 0 | 0 | s[] : | | | | | | | | | | | | | | | |
| === | === | === | === | === | === | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

end p1/0

5. Soma dos Cubos dos números naturais de 1 até 100(iterativamente)

a. P-code:

```

1  INT 0 3
2  LIT 0 1
3  STO 0 0
4  LIT 0 2
5  STO 0 1
6  LOD 0 1
7  LOD 0 1
8  OPR 0 4
9  LIT 0 2
10 OPR 0 4
11 LOD 0 0
12 OPR 0 2
13 STO 0 2
14 LOD 0 0
15 LOD 0 1
16 OPR 0 2
17 STO 0 1
18 LOD 0 1
19 LOD 0 1
20 OPR 0 4
21 LOD 0 1
22 OPR 0 4
23 LOD 0 2
24 OPR 0 2
25 STO 0 2
26 11
27 LOD 0 1
28 LIT 0 100
29 OPR 0 7
30 JPC 0 13
31 OPR 0 0

```

b. Saída do Código:

```

6  0 12 JPC 0 24 s[] : 0 0 0 100 328350 100 0
6  0 13 LOD 0 3 s[] : 0 0 0 100 328350 100 0 100
7  0 14 LOD 0 3 s[] : 0 0 0 100 328350 100 0 100 100
8  0 15 OPR 0 4 s[] : 0 0 0 100 328350 100 0 10000
7  0 16 LOD 0 4 s[] : 0 0 0 100 328350 100 0 10000 328350
8  0 17 OPR 0 2 s[] : 0 0 0 100 328350 100 0 338350
7  0 18 STO 0 4 s[] : 0 0 0 100 338350 100 0
6  0 19 LOD 0 3 s[] : 0 0 0 100 338350 100 0 100
7  0 20 LIT 0 1 s[] : 0 0 0 100 338350 100 0 100 1
8  0 21 OPR 0 2 s[] : 0 0 0 100 338350 100 0 101
7  0 22 STO 0 3 s[] : 0 0 0 101 338350 100 0
6  0 23 JMP 0 8 s[] : 0 0 0 101 338350 100 0
6  0 8 INT 0 -1 s[] : 0 0 0 101 338350 100
5  0 9 LOD 0 3 s[] : 0 0 0 101 338350 100 101
6  0 10 LOD 0 5 s[] : 0 0 0 101 338350 100 101 100
7  0 11 OPR 0 12 s[] : 0 0 0 101 338350 100 1 1
6  0 12 JPC 0 24 s[] : 0 0 0 101 338350 100
5  0 24 OPR 0 0 s[] :
=== === ===
t b p f l a 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
end pl/0

```


Link para download do código:

<https://github.com/LeoDNascimento/Compiladores/tree/main/Atividade1b>

Referências:

p-code machine. Disponível em: https://en.wikipedia.org/wiki/P-code_machine